(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2001 (25.10.2001)

PCT

(10) International Publication Number
WO 01/80483 A2

(51) International Patent Classification[7]: H04L 9/32

(21) International Application Number: PCT/US01/40507

(22) International Filing Date: 11 April 2001 (11.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/197,152	13 April 2000 (13.04.2000)	US
60/261,425	13 January 2001 (13.01.2001)	US
09/827,882	4 April 2001 (04.04.2001)	US

(71) Applicant *(for all designated States except US)*: BROADCOM CORPORATION [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).

(72) Inventors; and

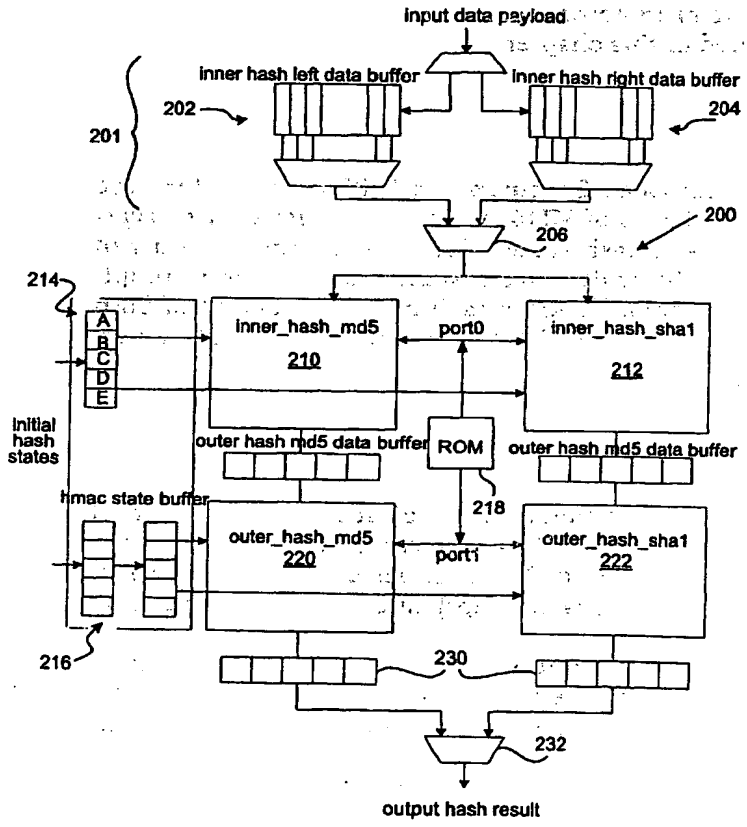
(75) Inventors/Applicants *(for US only)*: BUER, Mark [US/US]; 1027 E. Betsy Lane, Gilbert, AZ 85296 (US). LAW, Patrick, Y. [US/US]; 19 Jacklin Circle, Milpitas, CA 95035 (US). QI, Zheng [CN/US]; 13 Jacklin Circle, Milpitas, CA 95035 (US).

(74) Agent: AUSTIN, James, E.; Beyer Weaver & Thomas, LLP, P.O. Box 778, Berkeley, CA 94704-0778 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: AUTHENTICATION ENGINE ARCHITECTURE AND METHOD



(57) Abstract: Provided is an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network. Authentication engines in accordance with the present invention apply a variety of techniques that may include, in various applications, collapsing two multi-round authentication algorithm (e.g., SHA1 or MD5 or variants) processing rounds into one; reducing operational overhead by scheduling the additions required by a multi-round authentication algorithm in such a manner as to reduce the overall critical timing path ("hiding the ads"); and, for a multi-loop (e.g., HMAC) variant of a multi-round authentication algorithm, pipelining the inner and outer loops. In one particular example of applying the invention in an authentication engine using the HMAC-SHA1 algorithm of the IPSec protocol, collapsing of the conventional 80 SHA1 rounds into 40 rounds, hiding the ads, and pipelining the inner and outer loops allows HMAC-SHA1 to be conducted in approximately the same time as conventional SHA1.

WO 01/80483 A2

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations. refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

PATENT APPLICATION

AUTHENTICATION ENGINE ARCHITECTURE AND METHOD

BACKGROUND OF THE INVENTION

The present invention relates generally to the field of cryptography, and more specifically to an architecture and method for cryptography acceleration. In particular, the invention is directed to a hardware implementation to increase the speed at which authentication procedures may be performed on data packets transmitted over a computer network.

Many methods to perform cryptography are well known in the art and are discussed, for example, in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (1996, 2nd Edition), herein incorporated by reference. In order to improve the speed of cryptography processing, specialized cryptography accelerator chips have been developed. Cryptography accelerator chips may be included in routers or gateways, for example, in order to provide automatic IP packet encryption/decryption. By embedding cryptography functionality in network hardware, both system performance and data security are enhanced.

Cryptography protocols typically incorporate both encryption/decryption and authentication functionalities. Encryption/decryption relates to enciphering and deciphering data, authentication is concerned with data integrity, including confirming the identity of the transmitting party and ensuring that a data packet has not been

tampered with en route to the recipient. It is known that by incorporating both encryption and authentication functionalities in a single accelerator chip, over-all system performance can be enhanced.

Examples of cryptography protocols which incorporate encryption/decryption and authentication functionalities include SSL (Netscape Communications Corporation), commonly used in electronic commerce transactions, and the more recently promulgated industry security standard known as "IPSec." These protocols and their associated algorithms are well known in the cryptography art and are described in detail in National Institute of Standards and Technology (NIST), IETF and other specifications, some of which are identified (for example, by IETF RFC#) below for convenience. These specifications are incorporated herein by reference for all purposes.

SSL (v3) uses a variant of HMAC (RFC2104) for authentication. The underlying hash algorithm can be either MD5 (RFC1321) and SHA1 (NIST). In addition, the key generation algorithm in SSL also relies on a sequence of MD5 and SHA1 operations. SSL deploys algorithms such as RC4, DES, triple DES for encryption/decryption operations.

The IP layer security standard protocol, IPSec (RFC2406) specifies two standard algorithms for performing authentication operations, HMAC-MD5-96 (RFC2403) and HMAC-SHA1-96 (RFC2404). These algorithms are based on the underlying MD5 and SHA1 algorithms, respectively. The goal of the authentication computation is to generate a unique digital representation, called a digest, for the input data.

Both MD5 and SHA1 specify that data is to be processed in 512-bit blocks. If the data in a packet to be processed is not of a multiple of 512 bits, padding is applied to round up the data length to a multiple of 512 bits. Thus, if a data packet that is received by a chip for an authentication is larger then 512 bits, the packet is broken into 512-bits data blocks for authentication processing. If the packet is not a multiple of 512 bits, the data left over following splitting of the packet into complete 512-bit blocks must be padded in order to reach the 512-bit block processing size. The same is true if a packet contains fewer then 512 bits of data. For reference, a typical Ethernet packet is up to 1,500 bytes. When such a packet gets split into 512-bit blocks, only the last block gets padded and so that overall a relatively small percentage of padding overhead is required. However for shorter packets, the padding overhead can be much higher. For example, if a packet has just over 512 bits it will need to be divided into two 512-bit blocks, the second of which is mostly padding so that padding overhead approaches 50% of the process data. The authentication of such short data packets is particularly burdensome and time consuming using the conventionally implemented MD5 and SHA1 authentication algorithms.

For each 512-bit data block, a set of operations including non-linear functions, shift functions and additions, called a "round," is applied to the block repeatedly. MD5 and SHA1 specify 64 rounds and 80 rounds, respectively, based on different non-linear and shift functions, as well as different operating sequences. In every round, the operation starts with certain hash states (referred to as "context") held by hash state registers (in hardware) or variables (in software), and ends with a new set of hash states (i.e., an initial "set" of hash states and an end set; a "set" may be of 4 or 5 for the number of registers used by MD5 and SHA1, respectively). MD5 and SHA1

each specify a set of constants as the initial hash states for the first 512-bit block. The following blocks use initial hash states resulting from additions of the initial hash states and the ending hash states of the previous blocks.

Typically, MD5 and SHA1 rounds are translated into clock cycles in hardware implementations. The addition of the hash states, to the extent that they cannot be performed in parallel with other round operations, requires overhead clock cycles in the whole computation. The computation of the padded portion of the data is also generally considered performance overhead because it is not part of the true data. Accordingly, the performance of MD5 and SHA1 degrade the most when the length of the padding is about the same as the length of the data (e.g., as described above, when a packet has just fewer than 512 bits of data and the padding logic requires an extra 512-bit to be added for holding the pad values).

Moreover, the HMAC-MD5-96 and HMAC-SHA1-96 algorithms used in IPSec expand MD5 and SHA1, respectively, by performing two loops of operations. The HMAC algorithm for either MD5 or SHA1 (HMAC-x algorithm) is depicted in Fig. 1. The inner hash (inner loop) and the outer hash (outer loop) use different initial hash states. The outer hash is used to compute a digest based on the result of the inner hash. Since the result of inner hash is 128 bits long for MD5 and 160 bits long for SHA1, the result must always be padded up to 512 bits and the outer hash only processes the one 512-bit block of data. HMAC-MD5-96 and HMAC-SHA1-96 provide a higher level of security, however additional time is needed to perform the outer hash operation. This additional time becomes significant when the length of the data to be processed is short, in which case, the time required to perform the outer hash operation is comparable to the time required to perform the inner hash operation.

Authentication represents a significant proportion of the time required to complete cryptography operations in the application of cryptography protocols incorporating both encryption/decryption and MD5 and/or SHA1 authentication functionalities. In the case of IPSec, authentication is often the time limiting step, particularly for the processing or short packets, and thus creates a data processing bottleneck. Accordingly, techniques to accelerate authentication and relieve this bottleneck would be desirable. Further, accelerated implementations of multi-round authentication algorithms would benefit any application of these authentication algorithms.

SUMMARY OF THE INVENTION

In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network. As described in this application, the invention has particular application to the variants of the SHA1 and MD5 authentication algorithms specified by the IPSec cryptography standard. In accordance with the IPSec standard, the invention may be used in conjunction with data encryption/encryption architecture and protocols. However it is also suitable for use in conjunction with other non-IPSec cryptography algorithms, and for applications in which encryption/decryption is not conducted (in IPSec or not) and where it is purely authentication that is accelerated. Among other advantages, an authentication engine in accordance with the present invention provides improved performance with regard to the processing of short data packets.

Authentication engines in accordance with the present invention apply a variety of techniques that may include, in various applications, collapsing two multi-round authentication algorithm (e.g., SHA1 or MD5 or variants) processing rounds into one; reducing operational overhead by scheduling the additions required by a multi-round authentication algorithm in such a manner as to reduce the overall critical timing path ("hiding the adds"); and, for a multi-loop (e.g., HMAC) variant of a multi-round authentication algorithm, pipelining the inner and outer loops. In one particular example of applying the invention in an authentication engine using the HMAC-SHA1 algorithm of the IPSec protocol, collapsing of the conventional 80 SHA1 rounds into 40 rounds, hiding the adds, and pipelining the inner and outer loops

allows HMAC-SHA1 to be conducted in approximately the same time as conventional SHA1.

In one aspect, the present invention pertains to an authentication engine architecture for an multi-loop, multi-round authentication algorithm. The architecture includes a first instantiation of a multi-round authentication algorithm hash round logic in an inner hash engine, and a second instantiation of a multi-round authentication algorithm hash round logic in an outer hash engine. A dual-frame payload data input buffer configured for loading one new data block while another data block one is being processed in the inner hash engine, an initial hash state input buffer configuration for loading initial hash states to the inner and outer hash engines for concurrent inner hash and outer hash operations, and a dual-ported ROM configured for concurrent constant lookups for both inner and outer hash engines are also included. The multi-loop, multi-round authentication algorithm may be HMAC-MD5 or HMAC-SHA1.

In another aspect, the invention pertains to an authentication engine architecture for a multi-round authentication algorithm. The architecture includes a hash engine configured to implement hash round logic for a multi-round authentication algorithm. The hash round logic implementation included at least one addition module having a plurality of carry save adders for computation of partial products, and a carry look-ahead adder for computation and propagation of a final sum. The multi-round authentication algorithm may be MD5 or SHA1.

In another aspect, the invention pertains to an authentication engine architecture for an SHA1 authentication algorithm. The architecture includes at least

one hash engine configured to implement hash round logic. The logic implementation includes five hash state registers, one critical and four non-critical data paths associated with the five registers. In successive SHA1 rounds, registers having the critical path are alternative.

In another aspect, the invention pertains to a method of authenticating data transmitted over a computer network. The method involves receiving a data packet stream, splitting the packet data stream into fixed-size data blocks, and processing the fixed-size data blocks using a multi-loop, multi-round authentication engine architecture having a hash engine core with an inner hash engine and an outer hash engine. The architecture is configured to pipeline the hash operations of the inner hash and outer hash engines, collapse and rearrange multi-round logic to reduce rounds of hash operations, and implement multi-round logic to schedule addition computations to be conducted in parallel with round operations. The multi-loop, multi-round authentication algorithm may be HMAC-MD5 or HMAC-SHA1.

In another aspect, the invention pertains to a method of authenticating data transmitted over a computer network. The method involves receiving a data packet stream, splitting the packet data stream into fixed-size data blocks, processing the fixed-size data blocks using a multi-round authentication engine architecture. The architecture implements hash round logic for a multi-round authentication algorithm configured to schedule addition computations to be conducted in parallel with round operations. The multi-round authentication algorithm may be MD5 or SHA1.

In still another aspect, the invention pertains to a method of authenticating data transmitted over a computer network using an SHA1 authentication algorithm. The

method involves providing five hash state registers, and providing data paths from the five state registers such that four of the five data paths from the registers in any SHA1 round are not timing critical.

These and other features and advantages of the present invention will be presented in more detail in the following specification of the invention and the accompanying figures which illustrate by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

Fig. 1 is a high-level block diagram depicting the HMAC-x algorithm (HMAC for either MD5 or SHA1) implemented in the IPSec standard protocol.

Fig. 2 is a high-level block diagram of an authentication engine architecture in accordance with one embodiment the present invention.

Fig. 3 is a time study diagram illustrating the critical path of the conventional round logic of the SHA1 authentication algorithm.

Fig. 4 is a time study diagram illustrating the critical path of the round logic of the SHA1 authentication algorithm in accordance with one embodiment the present invention.

Fig. 5 is a high-level block diagram of an SHA1 hash engine illustrating the major elements of a round logic design in accordance with one embodiment the present invention.

Fig. 6 is a lower-level block diagram illustrating details of the scheduling of the additions within the round logic design of Fig. 5.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

Reference will now be made in detail to some specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

In general, the present invention provides an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network. Authentication engines in accordance with the present invention apply a variety of techniques that may include, in various applications, collapsing two multi-round authentication algorithm (e.g., SHA1 or MD5 or variants) processing rounds into one; reducing operational overhead by scheduling the additions required by a multi-round authentication algorithm (e.g., SHA1 or variants) in such a manner as to reduce the overall critical timing path ("hiding the adds"); and, for an HMAC (multi loop) variant of a multi-round

authentication algorithm, pipelining the inner and outer loops. Among other advantages, an authentication engine in accordance with the present invention provides improved performance with regard to the processing of short data packets.

In this specification and the appended claims, the singular forms "a," "an," and "the" include plural reference unless the context clearly dictates otherwise. Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood to one of ordinary skill in the art to which this invention belongs.

The present invention may be implemented in a variety of ways. As described in this application, the invention has particular application to the variants of the SHA1 and MD5 authentication algorithms specified by the IPSec cryptography standard. In the following description, the invention is discussed primarily in connection with the IPSec protocol. However, one of skill in the art will recognize that various aspects of the invention may also be applied to multi-loop and/or multi-round authentication algorithms generally, whether or not used with IPSec or in conjunction with cryptography operations at all. Further, while the aspects of the present invention described below are used together in a preferred embodiment of the invention, some aspects may be used independently to accelerate authentication operations. For example, the pipelining operations are particularly applicable to multi-loop, multi-round authentication algorithms; the round-collapsing operations are particularly applicable to SHA1 and variant authentication algorithms; while the scheduling of the additions may be applied to any multi-round authentication algorithm.

Pipelining Inner and Outer Hash Operations

Fig. 2 is a high-level block diagram of an authentication engine architecture in accordance with one embodiment the present invention. The engine architecture implements a pipelined structure to hide the time required for performing the outer hash operation when multiple data payloads are fed to the engine continuously. The engine architecture includes a core having two instantiations of the hash round logic; in this instance, inner and outer hash engines (inner and outer loops) for each of the MD5 hash round logic and the SHA1 hash round logic supported by the IPSec protocol. Pipeline control logic ensures that the outer hash operation for one data payload is performed in parallel with the inner hash operation of the next data payload in the packet stream fed to the authentication engine. A dual-frame input buffer is used for the inner hash engine, allowing one new 512-bit block to be loaded while another one is being processed, and the initial hash states are double buffered for concurrent inner hash and outer hash operations. In addition, dual-ported ROM is used for concurrent constant lookups by both inner and outer hash engines.

Referring to Fig. 2, the engine 200 includes a dual-frame input data payload buffer 201, in this instance having left frame 202, and a right frame 204. Input data payloads received by the engine 200, for example from data packets received off a network by a chip on which the engine architecture is implemented, are distributed between the frames 202, 204 of the input data buffer 201 so that one data block may be loaded into the buffer while another one is being processed downstream in the data flow. Since Fig. 2 illustrates an implementation of the present invention for processing IPSec packets, the architecture includes hash engines for the MD5 and SHA1 authentication protocols supported by IPSec. In accordance with the MD5 and SHA1 protocols, the input data payloads are loaded into the dual frames of the input

data buffer 201, split into 512-bit data blocks, padded if necessary (i.e., where the data block is less than 512 bits) and stored prior to being passed to an inner hash engine for processing. A multiplexer 206 controls the flow of 512-bit data blocks from the frames of the input buffer to an inner hash engine.

Initial hash states are needed on per packet basis for the first data block of each packet. Initial hash states, are generated by software based on the authentication key and some default constant states based on the HMAC algorithm (pre-hashed), in accordance the specifications for these algorithms. This is typically done once per key. Alternatively, the initial states may be derived from the default constant states and the authentication key using the same hardware for every packet that requires authentication.

The initial hash states for the inner hash of a given data block are loaded into a buffer 214 associated with the inner hash engine(s) 210, 212. The initial hash states for the outer hash of that data block are loaded into the first 215 of a pair of buffers 215, 216 (referred to as an HMAC state buffer) associated with the outer hash engine(s) 220, 222. When the initial hash states are passed to the inner hash engine for processing of the data block, the outer hash states for that block are loaded into the second buffer 216, and the inner and outer initial hash states for the next packet to be processed are loaded into the buffers 214, 215, respectively. In this way, the synchronization of the inner and outer hash states for a given data block is maintained, and the initial hash states are available for concurrent inner hash and outer hash operations. Further, the double buffering of the hash states allows initial hash states of the second packet to be loaded while the first packet is being processed so that the

data processing is continuous from packet to packet, thereby maximizing the efficiency and processing power of the hash engine.

The engine 200, further includes a dual-ported ROM 218. The dual-ported ROM 218 further facilitates the parallel inner and outer has operations by allowing for concurrent constant lookups by both inner and outer hash engines.

The inner hash is conducted on all 512 bit blocks of a given data packet. The result of inner hash is 128 bits long for MD5 and 160 bits long for SHA1. The result is padded up to 512 bits and the outer hash processes the one 512-bit block of data to compute a digest based on the result of the inner hash. An output buffer 230 stores the digest and outputs it through a multiplexer 232.

Collapsing Multi-Round Authentication Algorithm Processing Rounds

Of the two algorithms supported by the IPSEc protocol, HMAC-SHA1-96 is about twenty-five percent slower than HMAC-MD5-96 in terms of the total computation rounds. One way to improve HMAC-SHA1-96 in an IPSec-supporting hardware implementation is to collapse multiple rounds of logic into single clock cycle thus the total number of clocks required for HMAC-SHA1-96 operation is reduced. The same approach may be applied to any multi-round authentication algorithm. However, simply collapsing the logic for multiple rounds into a single clock cycle can cause the delay to compute the collapsed logic to increase, therefore reducing the maximum clock frequency.

Fig. 3 is a time study diagram illustrating the timing critical path of the conventional round logic of the SHA1 authentication algorithm. Registers a, b, c, d and e hold the intermediate hash states between rounds. They are duplicated in this

figure to demonstrate the ending points of the logic paths clearly. In the actual design, the paths are fed back to the same set of registers because the round logic is reused 80 times. The "+" symbols identify standard adders implemented as carry look-ahead adders (CLAs). W_i represents the incoming payload. K_i represents a constant, obtained from ROM used in the authentication computations. It is shown in the figure that the timing critical paths are from registers b, c and d, going through the non-linear function (defined by the SHA1 specification) and the adders and ending at register a. Registers b, c, d and e each receives a non-critical input (b receives a, etc.).

Fig. 4 is a time study diagram illustrating the timing critical path of the collapsed round logic of the SHA1 authentication algorithm in accordance with one embodiment the present invention. The SHA1 algorithm specifies five registers. As illustrated above, the data path of four of the five registers in any SHA1 round are not critical (time limiting). In accordance with this invention, in successive SHA1 rounds the registers having the critical path are alternative so that four registers worth of data may always be passed on to the next round prior to completion of the critical path in the current round. Thus, when two rounds of SHA1 are put together, the critical path computation of the second round is independent of that of the first round, since the receiving register of the critical path of the first round (i.e., register a) is not the driving register of the critical path of the second round (i.e., register e). This approach demonstrates how two SHA1 rounds may be collapsed together while maintaining the same amount of delay for the timing critical path, and how by alternating the critical path from register to register between rounds in this way, the adding operations may be "hidden."

In a preferred embodiment, the eighty rounds of an SHA1 loop are collapsed into forty rounds. As described and illustrated above, the collapsing of rounds is accomplished by having a single set of registers (the preferred embodiment has 5 registers as defined by the IPSec protocol) with two rounds of logic. It is contemplated that the techniques of invention described herein can also be applied to further collapse the number of SHA1 rounds in an SHA1 loop into twenty or even fewer rounds.

Scheduling the Additions

As described above, both MD5 and SHA1 algorithms specify that the final hash states of every 512-bit block to be added together with the initial hash states. The results are then used as the initial states of the next 512-bit block. In MD5, values of four pairs of 32-bit registers need to be added and in SHA1, five pairs. Considering that each 32-bit addition takes one clock cycle, a typical hardware implementation would use four extra cycles in MD5 and five extra cycles in SHA1 to perform these additions if hardware resources are limited.

As noted above with reference to Figs. 3 and 4, in both MD5 and SHA1, only one state register is re-computed every round. The rest of the state registers use shifted or non-shifted contents from neighboring registers. Thus, the final hash states are not generated in the final round, but rather in the last four consecutive MD5 rounds or five SHA1 rounds, respectively. The present invention exploits this observation by providing architecture and logic enabling the scheduling of the additions as early as the final hash state is available, hiding the computation time completely behind the round operations. This is illustrated in the following

scheduling tables in which 'Ti' represents one clock cycle and 'rnd i' represents round operation. The initial hash states are represented by ia, ib, ic, id and ie. Parallel operations are listed in the same column.

MD5

T1	T2	T3	ooooooooooo	T61	T62	T63	T64	T1
rnd 1	rnd 2	rnd 3		rnd 61	rnd 62	rnd 63	rnd 64	rnd 1
					a+ia	d+id	c+ic	b+ib

original SHA1

T1	T2	T3	ooooooooooo	T77	T78	T79	T80	T1
rnd 1	rnd 2	rnd 3		rnd 77	rnd 78	rnd 79	rnd 80	rnd 1
				e+ie	d+id	c+ic	b+ib	a+ia

collapsed SHA1

T1	T2	T3	ooooooooooo	T38	T39	T40	T1
rnd 1	rnd 2	rnd 3		rnd 38	rnd 39	rnd 40	rnd 1
					e+ie	d+id c+ic	b+ib a+ia

In one embodiment of the invention, a plurality of adds with the final hash states may be accomplished in a single clock cycle. An example is shown in the "collapsed SHA1" table, in which the five adds are performed in just three clock cycles T39, T40 and T1 of the next loop. One of skill in the art will recognize that, consistent with the principles of this invention described herein, it is possible to perform more than two adds in parallel in one clock cycle. Moreover, it should be noted that, as illustrated in the tables, this aspect of the present invention is applicable

to both collapsed and non-collapsed multi-round authentication algorithms. Implementation of this aspect of the present invention in conjunction with a collapsed multi-round algorithm is particularly advantageous since hiding of adding steps becomes increasingly important as the number of rounds is decreased. Adds that are not hidden in the manner of this aspect of the present invention would represent an even larger proportion of overhead in a collapsed round implementation than in an implementation with a higher number of rounds.

Logic Design

Fig. 5 is a high-level block diagram of an SHA1 hash engine illustrating the major elements of a collapsed round logic design in accordance with one embodiment the present invention consistent with the timing critical path study of Fig. 4. The design makes use of carry save adders (CSA; delay is equivalent to 1-bit adder), taking advantage of their capacity to add multiple quantities together. CSAs efficiently add multiple quantities together to generate partial products which are not propagated. Two comprehensive addition modules, add5to1 and add4to1 in the figure each uses several stages of CSA followed-by a carry look-ahead (CLA) adder, as illustrated and described in more detail with reference to Fig. 6, below.

The hash engine has five registers, A, B, C, D and E. The initial hash state in register A (a_1) goes through a 5-bit circular shift and is added to the initial hash state in register E (e_1), the payload data (W_i), a constant (K_i), and the result of a function (F_t) of the initial hash states in registers B, C and D by an add5to1 adder module that is built by CSA and CLA adders. The initial hash state in register D (d_1) is added to the payload data (W_{i+1}), a constant (K_{i+1}), and the result of a function (F_t) of the initial

hash states in registers A, B (which passes through a 30-bit circular shift) and C by an add4to1 adder module that is built by CSA and CLA adders.

The adder modules conclude with a carry look-ahead (CLA) adder. The sum of each adder module is added by a CLA adder to generate and propagate a final sum for the round which is then fed back into register A for the next round. The most timing critical input of these two modules needs only to go through the last CLA stage.

Fig. 6 is a lower-level block diagram illustrating details of the scheduling of the additions within the round logic design of Fig. 5. Unrolling two rounds of SHA1 operation will lead to a speed path of:

$$S = ((a<<<5) + f(b, c, d)+ e + w + k)<<<5+ f(b, c, d) + e + w + k,$$

where, a, b, c, d, e, w and k are 32-bit quantities. In accordance with the embodiment of the present invention depicted in Fig. 5, the operation is done in two steps. Step 1 uses module add5to1 to generate:

$$Si = (a<<<5) + f(b, c, d)+ e + w + k.$$

Step 2 uses module add4to1 and a 32-bit carry look-ahead adder (CLA) to generate:

$$S = Si<<<5+ f(b, c, d) + e + w + k.$$

In each step, carry save adders (CSA) are used to perform 3-2 input reduction before the 32-bit CLA is applied. The overall delay is equivalent to two 32-bit CLA delays plus one 32-bit CSA delay plus the delay for function 'f' for the most timing critical path. After all the reductions are completed via CSAs, Step 1 and Step 2 become:

$S = (A+B)<<<5+C+D.$

Implementations of the invention using this logic design in an authentication engine using the HMAC-SHA1 algorithm of the IPSec protocol, collapsing of the conventional 80 SHA1 rounds into 40 rounds, hiding the adds, and pipelining the inner and outer loops have enabled HMAC-SHA1 to be conducted in approximately the same time as conventional SHA1.

Conclusion

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, those skilled in the art will appreciate that various adaptations and modifications of the just-described preferred embodiments can be configured without departing from the scope and spirit of the invention. For example, while the present invention has been described primarily in connection with the IPSec protocol, the principles of the invention may also be applied to multi-round authentication algorithms generally, whether or not used in conjunction with cryptography operations. Therefore, the described embodiments should be taken as illustrative and not restrictive, and the invention should not be limited to the details given herein but should be defined by the following claims and their full scope of equivalents.

What is claimed is:

CLAIMS

1. An authentication engine architecture for an multi-loop, multi-round authentication algorithm, comprising:

a first instantiation of a multi-round authentication algorithm hash round logic in an inner hash engine;

a second instantiation of a multi-round authentication algorithm hash round logic in an outer hash engine;

a dual-frame payload data input buffer configured for loading one new data block while another data block one is being processed in the inner hash engine;

an initial hash state input buffer configuration for loading initial hash states to the inner and outer hash engines for concurrent inner hash and outer hash operations; and

a dual-ported ROM configured for concurrent constant lookups for both inner and outer hash engines.

2. The authentication engine architecture of claim 1, wherein the multi-loop, multi-round authentication algorithm is HMAC-MD5.

3. The authentication engine architecture of claim 1, wherein the multi-loop, multi-round authentication algorithm is HMAC-SHA1.

4. The authentication engine architecture of claim 1, wherein at least one of the inner and outer hash engines is configured to implement hash round logic including at least one addition module comprising:

a plurality of carry save adders for computation of partial products; and

a carry look-ahead adder for computation and propagation of a final sum.

5. The authentication engine of claim 4, wherein the carry save adders and the carry look-ahead adder are configured such that addition computations are conducted in parallel with round operations.

6. The authentication engine architecture of claim 3, wherein at least one of the inner and outer hash engines is configured to implement hash round logic comprising:

five hash state registers;

one critical and four non-critical data paths associated with the five registers, such that in successive SHA1 rounds, registers having the critical path are alternative.

7. The authentication engine architecture of claim 6, wherein said hash round logic is implemented such that eighty rounds of an SHA1 loop are collapsed into forty rounds.

8. The authentication engine architecture of claim 3, wherein at least one of the inner and outer hash engines is configured to implement hash round logic comprising:

five hash state registers;

a 5-bit circular shifter;

an add5to1 adder module having a plurality of CSAs and a CLA adder;

a 30-bit circular shifter; and

an add4to1 adder module having a plurality of CSAs and a CLA adder.

9. An authentication engine architecture for a multi-round authentication algorithm, comprising:

a hash engine configured to implement hash round logic for a multi-round authentication algorithm, said hash round logic implementation including at least one addition module comprising,

a plurality of carry save adders for computation of partial products, and

a carry look-ahead adder for computation and propagation of a final sum.

10. The authentication engine of claim 9, wherein the carry save adders and the carry look-ahead adder are configured such that addition computations are conducted in parallel with round operations.

11. The authentication engine architecture of claim 9, wherein the multi-round authentication algorithm is MD5.

12. The authentication engine architecture of claim 9, wherein the multi-round authentication algorithm is SHA1.

13. The authentication engine architecture of claim 12, wherein the hash round logic implementation comprises:

five hash state registers;

a 5-bit circular shifter;

an add5to1 adder module having a plurality of CSAs and a CLA adder;

a 30-bit circular shifter; and

an add4to1 adder module having a plurality of CSAs and a CLA adder.

14. An authentication engine architecture for an SHA1 authentication algorithm, comprising:

at least one hash engine configured to implement hash round logic comprising:

five hash state registers;

one critical and four non-critical data paths associated with the five registers, such that in successive SHA1 rounds, registers having the critical path are alternative.

15. The authentication engine architecture of claim 14, wherein said hash round logic is implemented such that eighty rounds of an SHA1 loop are collapsed into forty rounds.

16. A method of authenticating data transmitted over a computer network, comprising:

receiving a data packet stream;

splitting the packet data stream into fixed-size data blocks; and

processing the fixed-size data blocks using a multi-loop, multi-round authentication engine architecture having a hash engine core comprising an inner hash engine and an outer hash engine, said architecture configured to,

pipeline hash operations of said inner hash and outer hash engines,

collapse and rearrange multi-round logic to reduce rounds of hash operations, and

implement multi-round logic to schedule addition computations to be conducted in parallel with round operations.

17. The method of claim 16, wherein said pipelining comprises performance of an outer hash operation for one data payload in parallel with an inner hash operation of a second data payload in a packet stream fed to the authentication engine.

18. The method of claim 17, wherein a dual-frame input buffer is used for the inner hash engine.

19. The method of claim 18, wherein initial hash states for the hash operations are double buffered for concurrent inner hash and outer hash operations.

20. The method of claim 19, wherein concurrent constant lookups are performed from a dual-ported ROM by both inner and outer hash engines.

21. The method of claim 16, wherein the multi-loop, multi-round authentication algorithm is MD5.

22. The method of claim 16, wherein the multi-loop, multi-round authentication algorithm is SHA1.

23. The method of claim 22 wherein said scheduling of additions comprises:

conducting a 5-bit circular shift on data from a first register;

adding an initial hash state in a second register, a first payload data block, a first constant, and the result of a function (F_t) of the initial hash states in third, fourth and fifth additional registers with an add5to1 adder module having a plurality of CSAs and a CLA adder;

conducting a 30-bit circular shift on data from the third additional register; and

adding the initial hash state in the fourth additional register to a second payload block, a second constant, and the result of a function (F_t) of the initial hash states in the first and fifth registers and the shifted hash state of the third register with an add4to1 adder module having a plurality of CSAs and a CLA adder.

24. The method of claim 22, wherein said collapsing and rearranging of the multi-round logic comprises:

providing five hash state registers; and

providing data paths from said five state registers such that four of the five data paths from the registers in any SHA1 round are not timing critical.

25. The method of claim 24, wherein, in successive SHA1 rounds, registers having the critical path are alternative.

26. The method of claim 25, wherein eighty rounds of an SHA1 loop are collapsed into forty rounds.

27. A method of authenticating data transmitted over a computer network, comprising:

receiving a data packet stream;

splitting the packet data stream into fixed-size data blocks; and

processing the fixed-size data blocks using a multi-round authentication engine architecture, said architecture implementing hash round logic for a multi-round authentication algorithm configured to schedule addition computations to be conducted in parallel with round operations.

28. The method of claim 27 wherein said hash round logic comprises:

conducting a 5-bit circular shift on data from a first register;

adding an initial hash state in a second register, a first payload data block, a first constant, and the result of a function (F_t) of the initial hash states in third, fourth and fifth additional registers with an add5to1 adder module having a plurality of CSAs and a CLA adder;

conducting a 30-bit circular shift on data from the third additional register; and

adding the initial hash state in the fourth additional register to a second payload block, a second constant, and the result of a function (F_t) of the initial hash states in the first and fifth registers and the shifted hash state of the third register with an add4to1 adder module having a plurality of CSAs and a CLA adder.

29. A method of authenticating data transmitted over a computer network using an SHA1 authentication algorithm, comprising:

providing five hash state registers; and

providing data paths from said five state registers such that four of the five data paths from the registers in any SHA1 round are not timing critical.

30. The method of claim 29, wherein, in successive SHA1 rounds, registers having the critical path are alternative.

31. The method of claim 30, wherein eighty rounds of an SHA1 loop are collapsed into forty rounds.

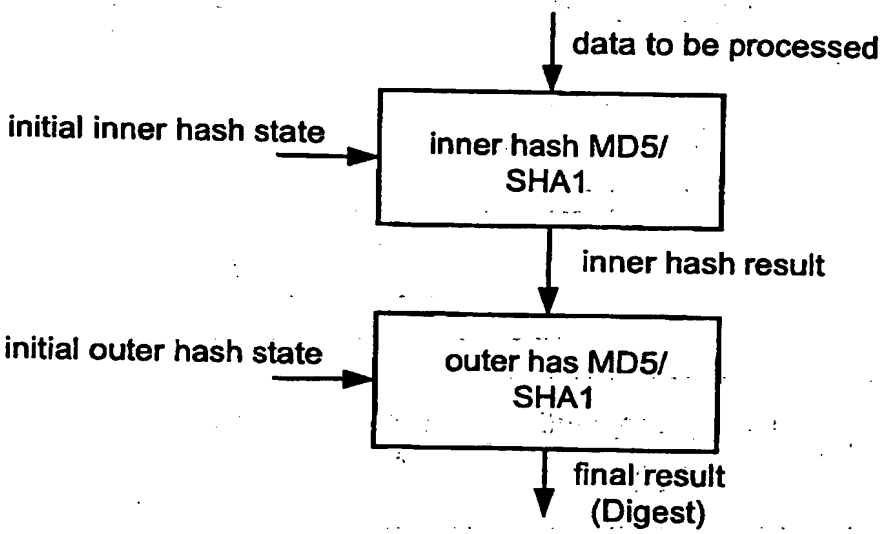


FIG. 1

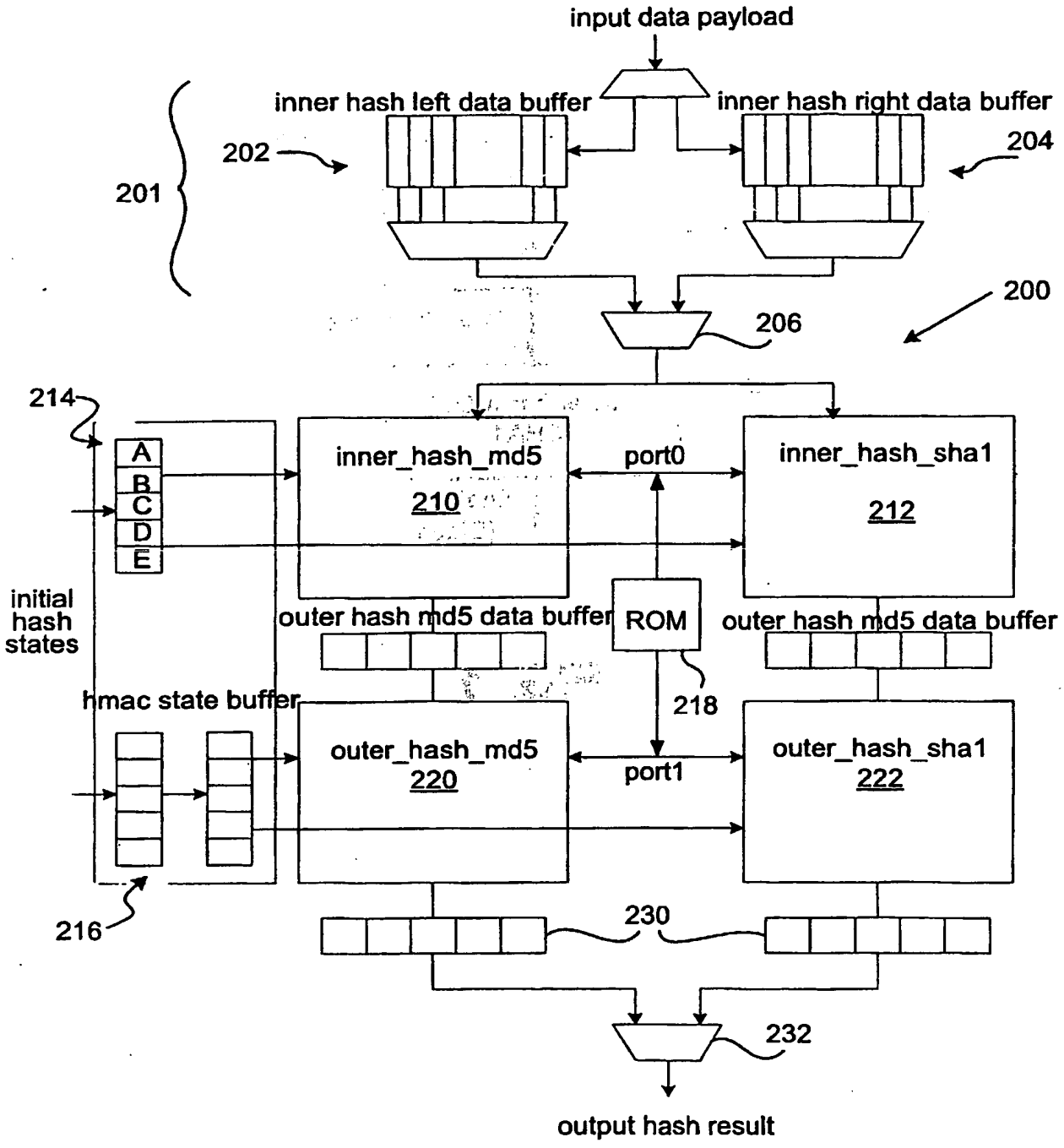

input data payload
inner hash left data buffer
inner hash right data buffer
202
204
201
200
206
214
A
B
C
D
E
inner_hash_md5
210
port0
inner_hash_sha1
212
initial
hash
states
outer hash md5 data buffer
ROM
outer hash md5 data buffer
218
hmac state buffer
outer_hash_md5
220
port1
outer_hash_sha1
222
216
230
232
output hash result

FIG. 2

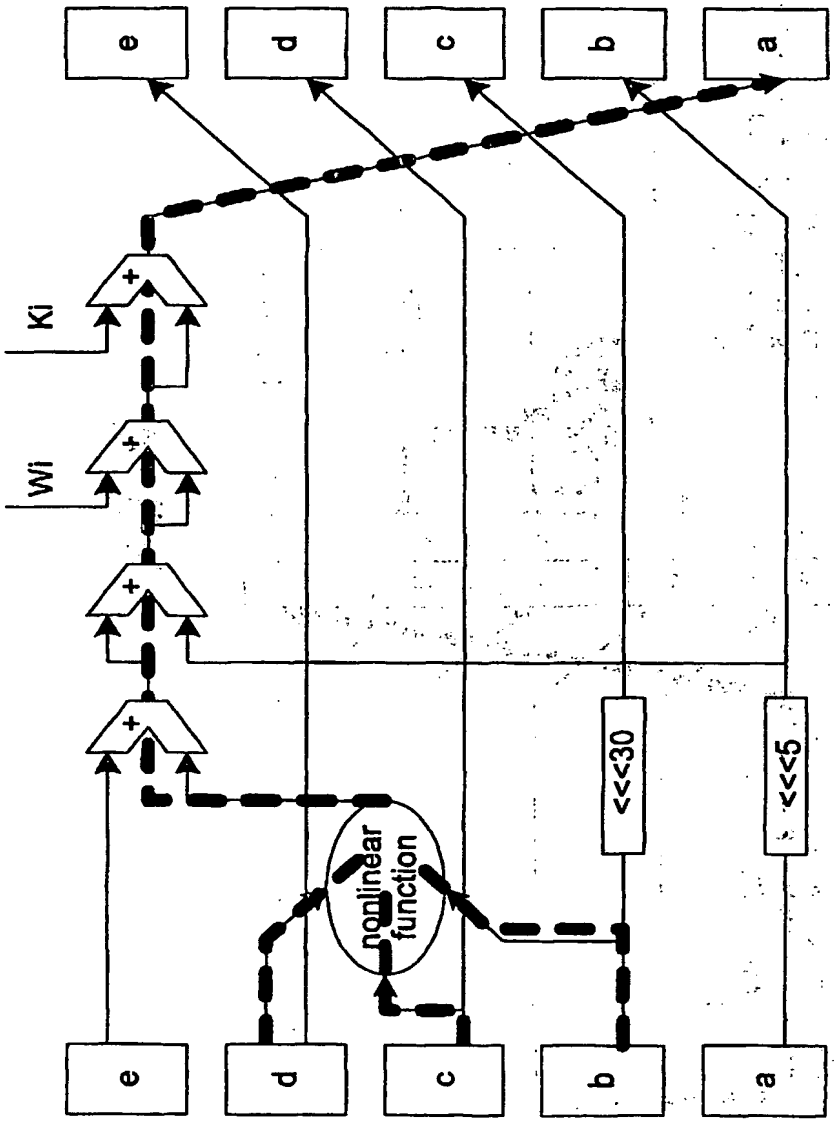


FIG. 3

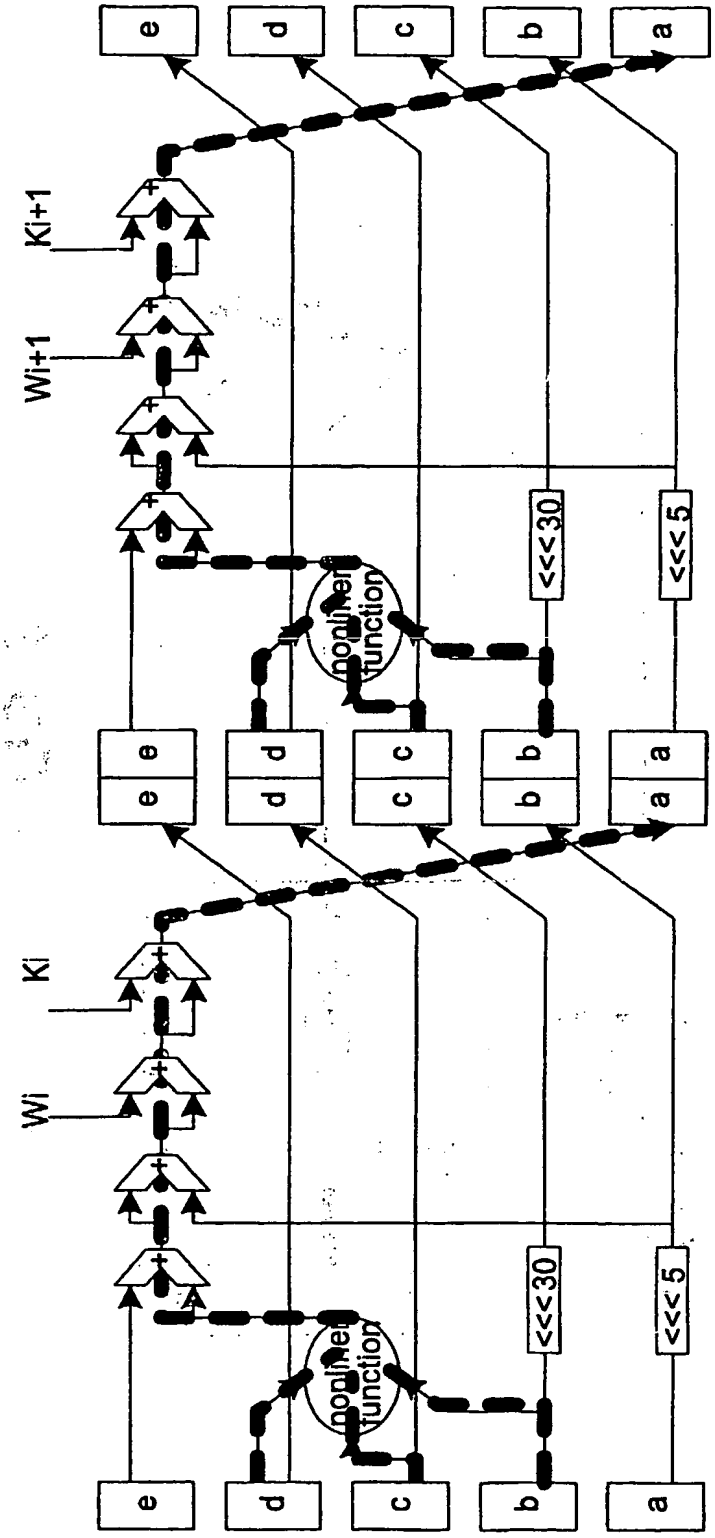


FIG. 4

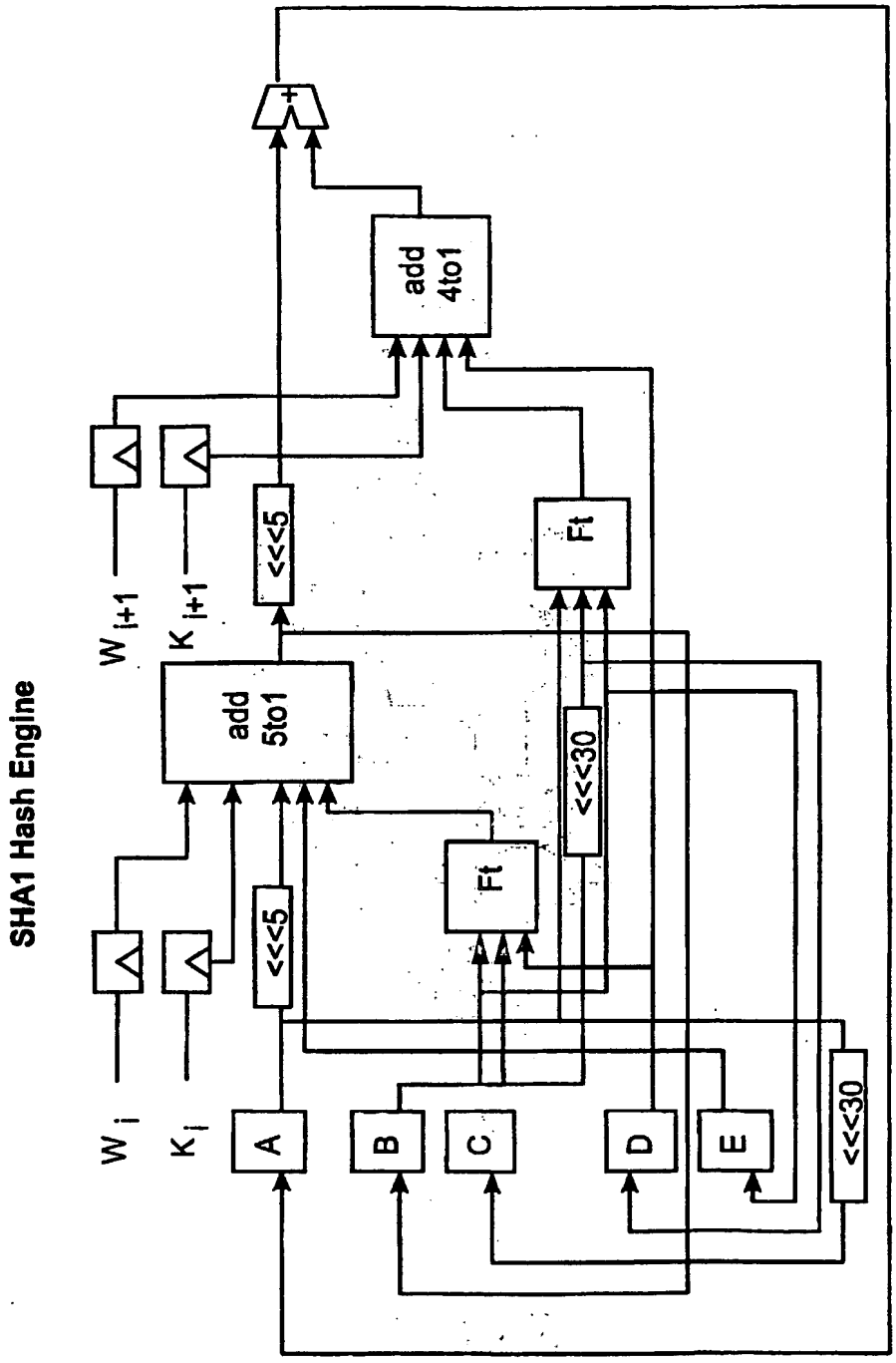


FIG. 5

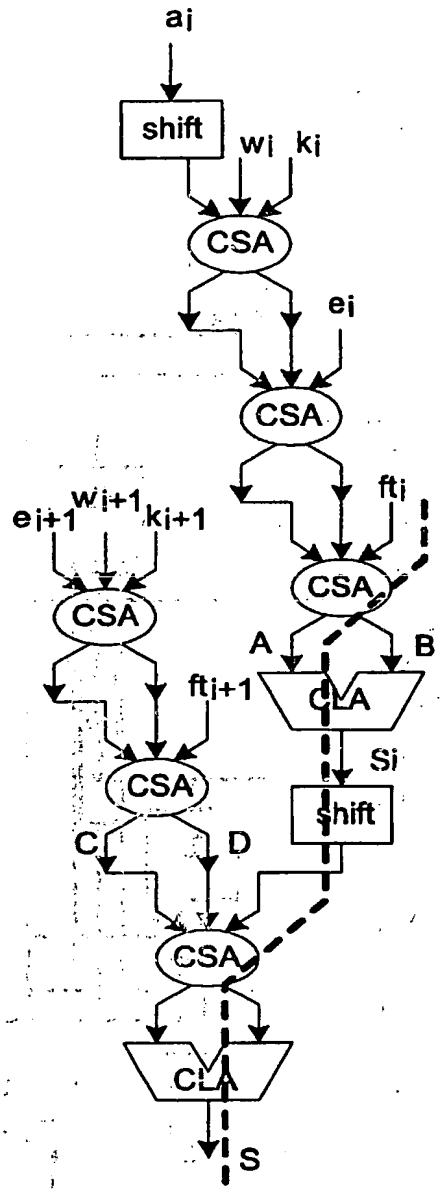


FIG. 6

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 October 2001 (25.10.2001)

PCT

(10) International Publication Number
WO 01/80483 A3

(51) International Patent Classification[7]: H04L 9/32

(21) International Application Number: PCT/US01/40507

(22) International Filing Date: 11 April 2001 (11.04.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/197,152	13 April 2000 (13.04.2000)	US
60/261,425	13 January 2001 (13.01.2001)	US
09/827,882	4 April 2001 (04.04.2001)	US

(71) Applicant *(for all designated States except US)*: **BROADCOM CORPORATION** [US/US]; 16215 Alton Parkway, Irvine, CA 92618-3616 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BUER, Mark** [US/US]; 1027 E. Betsy Lane, Gilbert, AZ 85296 (US). **LAW, Patrick, Y.** [US/US]; 19 Jacklin Circle, Milpitas, CA 95035 (US). **QI, Zheng** [CN/US]; 13 Jacklin Circle, Milpitas, CA 95035 (US).

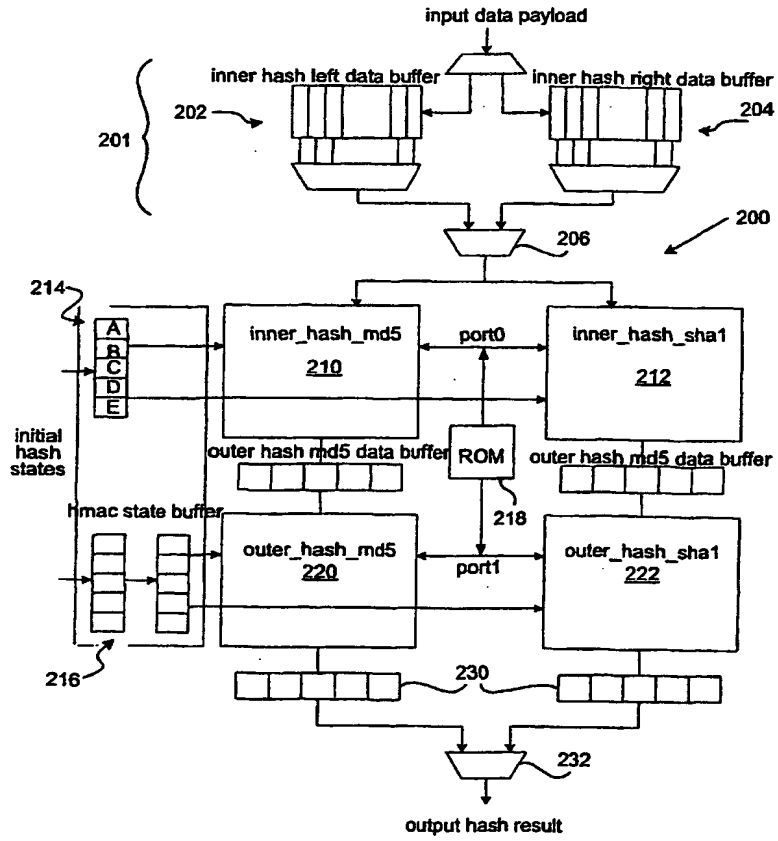
(74) Agent: **AUSTIN, James, E.**; Beyer Weaver & Thomas, LLP, P.O. Box 778, Berkeley, CA 94704-0778 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: AUTHENTICATION ENGINE ARCHITECTURE AND METHOD



(57) Abstract: Provided is an architecture (hardware implementation) for an authentication engine to increase the speed at which multi-loop and/or multi-round authentication algorithms may be performed on data packets transmitted over a computer network. Authentication engines in accordance with the present invention apply a variety of techniques that may include, in various applications, collapsing two multi-round authentication algorithm (e.g., SHA1 or MD5 or variants) processing rounds into one; reducing operational overhead by scheduling the additions required by a multi-round authentication algorithm in such a manner as to reduce the overall critical timing path ("hiding the ads"); and, for a multi-loop (e.g., HMAC) variant of a multi-round authentication algorithm, pipelining the inner and outer loops. In one particular example of applying the invention in an authentication engine using the HMAC-SHA1 algorithm of the IPSec protocol, collapsing of the conventional 80 SHA1 rounds into 40 rounds, hiding the ads, and pipelining the inner and outer loops allows HMAC-SHA1 to be conducted in approximately the same time as conventional SHA1.

WO 01/80483 A3

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR). OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
4 April 2002

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 01/40507

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, PAJ, EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SCHNEIER B: "APPLIED CRYPTOGRAPHY, SECOND EDITION" 1996 , JOHN WILEY & SONS , NEW YORK US XP002184521 cited in the application	14,29,30
A	page 436, paragraph 18.5 -page 440 page 442, paragraph 18.7 -page 444	1-3,6,9, 11,12, 16,21, 22,24, 25,29

-/--

[X] Further documents are listed in the continuation of box C.

[ ] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 December 2001	20/12/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl. Fax: (+31-70) 340-3016	Masche, C

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int[illegible]ional Application No
PC[illegible]/US 01/40507

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	TOUCH J D: "PERFORMANCE ANALYSIS OF MD5" COMPUTER COMMUNICATIONS REVIEW, ASSOCIATION FOR COMPUTING MACHINERY. NEW YORK, US, vol. 25, no. 4, 1 October 1995 (1995-10-01), pages 77-86, XP000541653 ISSN: 0146-4833 figures 2,7-10 page 84, right-hand column, line 11 - line 21 page 84, left-hand column, line 12 page 83, left-hand column, line 15 -right-hand column, line 7 abstract	27
A	---	1,2,5,8, 9,11,13, 16,21, 23,28
A	BELLARE M: "MESSAGE AUTHENTICATION USING HASH FUNCTIONS - THE HMAC CONSTRUCTION" RSA LABORATORIES' CRYPTOBYTES, vol. 2, no. 1, 1996, pages 1-5, XP002184520 abstract page 2, left-hand column, line 17 -right-hand column, line 31	1-3,9, 11,12, 14,16, 21,22,29
A	--- STALLINGS W: "SHA: THE SECURE HASH ALGORITHM PUTTING MESSAGE DIGESTS TO WORK" DR. DOBBS JOURNAL, REDWOOD CITY, CA, US, 1 April 1994 (1994-04-01), page 32,34 XP000570561 the whole document	1,3,6,9, 12,14, 16,22, 27,29

1